Optimizing Queries Executed on a Heterogeneous Spark Cluster

Iman Elghandour
Université Libre de Bruxelles
ielghand@ulb.ac.be

The recent advancement in technology has enabled researchers to capture huge data sets that are useful for various applications such as astronomy image processing, oil and gas mining, and meteorological data analysis. Analyzing such big data, which is usually compute-intensive, can reveal important insights for stakeholders and decision makers. To efficiently process such applications, many solutions have been proposed including: (1) high-throughput computing (HTC) systems such as Condor; (2) parallel computing systems leveraging efficient compute resources such as GPGPUs; and (3) distributed computing platforms such as MapReduce and Spark. However, each of these approaches has its own limitations. The HTC systems provide a limited programming interface flexibility and require professional tuning for each application. Parallel programming frameworks that leverage accelerators such as GPUs achieve very high parallelism. However, they are not suitable for dealing with large data sets because of the limited size of the on-chip memory of the accelerators and the high overhead introduced by transferring data between the host and the accelerator and vice-versa. On the contrary, the current available implementations of distributed computing platforms such as Hadoop and Spark have been efficiently used for processing very large data sets. However, the parallelism in Hadoop and Spark is limited by the number of CPU cores available at each node in the cluster.

Executing applications on a cluster equipped with heterogeneous resources such as various types of accelerators and processors can significantly improve the performance of the applications compared to running them on homogeneous clusters that are only equipped with CPUs. Additionally, using a distributed computing platform such as Hadoop or Spark to manage the cluster has benefits such as flexibility, transparency, reliability, and scalability. In this paper, we propose a platform that leverages Spark to manage a heterogeneous cluster of machines that are equipped with GPGPUs in addition to CPUs. In our earlier work [1], we have developed SparkGPU, an extension to Apache Spark that executes its input applications and SQL queries on a heterogeneous cluster that has nodes equipped with CPUs and GPUs. SparkGPU employs a node manager at each worker node to perform scheduling of tasks assigned to that worker to run on its CPU cores and GPUs. Our objective is to efficiently use all the computing resources at each node of the cluster to enhance the execution time of Spark jobs. We introduce a machine learning model that predicts the execution time of spark jobs/tasks on GPUs. The node manager uses the predicted execution times of input spark tasks when executed on CPUs and GPUs to efficiently schedule these tasks on a cluster node. We show that SparkGPU is capable of predicting the execution time of several machine learning applications using gradient boosting regression with an accuracy of 76%. We also show that SparkGPUs can process a compute intensive astronomical application in real-time. This is a speedup of two orders of magnitude as compared to a sequential program implementation for the same application.

References

[1] M. Hassaan and I. Elghandour. A real-time big data analysis framework on a CPU/GPU heterogeneous cluster: A meteorological application case study. In *IEEE/ACM Int. Conf. on Big Data Computing, Applications, and Technologies (BDCAT)*, pages 168–177, 2016.